

understood, however, that the usage information could be reported via any other suitable connection.

The present invention returns usage information to a web site (or other location) on a per channel basis either by proxy (which requires no change to web site operation) or by log files (which requires no change to web site operation other than simply acquiring and processing the log files). With proxy reporting, the cache hit tracker 40 preferably operates (from the web servers' perspective) as an HTTP proxy server. The cache hit tracker 40 stores reported "hits" from the content viewer 58 in the receiver 26 and subsequently performs a HTTP operation for each reported URL. This minimizes the time that the receiver 26 needs to be connected to the Internet 14 as it is not delayed while its hits are reported to the web servers 10. The type of HTTP operation performed by the cache hit tracker 40 is defined by the channel definitions, which are sent along with the usage reports 42 to the cache hit tracker 40. Specifically, the following HTTP operations may be configured to perform proxy usage reporting on a per channel basis:

(1) GET - The cache hit tracker 40 may perform an HTTP GET operation (which, generally, transfers URL data item from the web server 10 to the cache hit tracker 40.) The cache hit tracker 40 then waits for all the data to be received, but discards the data as it is received. This operation most closely matches what the web server 10 would have received had the user directly retrieved the URL data item without use of the present invention.

(2) GET no wait - The cache hit tracker 40 may perform an HTTP GET operation (just as above) but waits only for the HTTP response header to be received from the web server 10. Once received, the cache hit tracker 40 immediately closes the connection. This operation uses less network bandwidth, as the actual URL data item need not be completely transferred.

(3) GET IfModified Since - The cache hit tracker 40 may perform a GET If-Modified-Since HTTP operation, wherein the last-modified data comes from the HTTP response header included with the URL in the package. This